

Importancia del pensamiento abstracto. Su formación en el aprendizaje de la Programación

Importance of abstract thinking. His training in learning programming

Juan Carlos Fonden Calzadilla

Universidad Tecnológica de la Habana “José Antonio Echeverría”, Cuba.

Correo(s) electrónico(s):

fonden1980@gmail.com

fonden@ind.cujae.edu.cu.

Recibido: 9 de enero de 2019Aceptado: 12 de octubre de 2019

Resumen: El presente trabajo, encaminado a profesores de programación, programadores, estudiantes de Ingeniería y Ciencias de la Computación, argumenta la importancia de la formación del pensamiento abstracto a través del Proceso de Enseñanza Aprendizaje de la Programación. Empleando los métodos de investigación científica: análisis documental, modelación, enfoque de sistema, observación, las tormentas de ideas unido a la experiencia del autor como profesor de programación en diferentes contextos universitarios, se identificó un conjunto de recomendaciones didácticas y metodológicas que propician el desarrollo la abstracción en los estudiantes a través del aprendizaje de la Programación Orientada a Objetos y las Estructuras de Datos.

Palabras Clave: Pensamiento Abstracto, Recomendaciones Didácticas, Programación Orientada a Objetos, Estructuras de Datos

Abstract: The present work, directed to professors of programming, programmers, students of Engineering and Sciences of the Computation, argues the importance of the formation of the abstract thought through the Process of Teaching Learning of the Programming. Using the methods of scientific research: documentary analysis, modeling, system approach, observation, storms of ideas together with the author's experience as a professor of programming in different university contexts, a set of didactic and methodological recommendations was identified that favor the development the abstraction in the students through the learning of the Programming Oriented to Objects and the Data Structures.

Keywords: Abstract Thinking, Didactic Recommendations, Object-Oriented Programming, Data Structures

Introducción

Es una facultad del ser humano la capacidad de crear ideas a partir de otras, imaginar sucesos, narrar historias, modelar procesos, leer e interpretar problemas, planificar soluciones, proyectarse al futuro y soñar, aunque a muchas personas les puede ser difícil desarrollar estas capacidades.

El acto de poder aislar imaginariamente un conjunto de rasgos de un objeto o proceso para concentrarse en otros que deben ser examinados y luego volver representarse íntegramente el objeto original, con todas sus partes integrante y las relaciones entre ellas, es un proceso u operación del pensamiento que se conoce como abstracción.

Es también abstracción, sumergirse en el recuerdo, en una especie de viaje mental en el tiempo, y recordar acontecimientos pasados, provechosos o no, para imaginar el futuro, así como concentrarse y representar en la mente los textos, dibujos, tablas y fórmulas de un libro leído.

El pensamiento abstracto posibilita penetrar con profundidad en los pormenores de las cosas que examinamos y escudriñar más allá de lo que se puede advertir solamente con las informaciones adquiridas a través de los órganos sensoriales. De esta forma se alcanzan soluciones creativas a problemas con un alto grado de dificultad (Naranjo y Puga, 2016).

Han expresado Flores (2011) que *“El pensamiento abstracto es fundamental en la informática y la tecnología para comprender el cuerpo principal del problema de los computadores. Pensar en abstracto es una interesante heurística de propósito muy general que puede ayudar a enfrentar la solución de un problema.”* (p.2). El pensamiento abstracto es el reflejo indirecto y generalizado del contexto que envuelve una persona, es un modo de descubrir el mundo más allá de los datos e informaciones que se adquieren a través de los sentidos.

Por otra parte otros autores consideran que la abstracción es un proceso que se produce en la mente humana para separar detalles, con el objetivo de centrarse en lo verdaderamente significativo del problema para crear un modelo abstracto de la solución, (Serna y Polo, 2014) y que en el razonamiento abstracto, la persona no necesita del uso del lenguaje para solucionar una situación problemática sino que lo fundamental está en comprender las ideas por medio del razonamiento práctico o visual, mediante un lenguaje interiorizado.

Al emplear el pensamiento abstracto, la persona consigue imaginar, extrapolar lo asimilado a diferentes escenarios, construir proyectos, contextualizarse en diferentes tiempos y situaciones, predecir el futuro, viajar al pasado, crear ideas, realizar la comprensión adecuada de un problema, obtener conclusiones acertadas, crear hipótesis, definir conceptos y comprobar realidades.

La importancia de desarrollar el pensamiento abstracto.

Tanto en el diario vivir como en los diversos campos del saber, la abstracción es una operación básica del pensamiento humano, indispensable para el desarrollo de altos niveles intelectuales y

científicos ya que posibilita la realización de resúmenes, comparaciones, clasificar objetos y procesos y generar nuevas ideas para resolver problemas.

La sistematización de los estudios realizados por el autor en diversas publicaciones les permite identificar las siguientes ventajas del pensamiento abstracto:

1. Hace que los desafíos parezcan más realizables
2. Contribuye a ser más persuasivos con los demás al tratar un tema
3. Profundiza más en la esencia de un asunto y no se queda en la superficie del conocimiento
4. Hace al individuo más autónomo, creativo, inteligente, amable y empático con los demás.
5. La abstracción además de ser una operación del pensamiento es tratado también como un método de investigación científica.
6. El pensamiento abstracto es fundamental en las artes plásticas, la música, la oratoria y la enseñanza de la Programación.
7. El pensamiento abstracto es esencial en sentido general, para la modelación de objetos y procesos.

Rosental y Ludin (1973) plantearon que *“Todo conocimiento se halla necesariamente unido a procesos de abstracción. Sin ellos no sería posible descubrir la esencia del objeto, penetrar en su profundidad. La división del objeto en partes y la eliminación de las que nos son esenciales en él”* (p.2).

La abstracción en el paradigma de la Programación Orientada a Objetos.

La abstracción es un término del diario vivir aplicado al contexto de la Programación Orientada a Objetos (POO). Como ya se ha expresado, lo abstracto está en el ámbito de las ideas, los pensamientos, que no se puede concretar en algo material o tangible. Por ejemplo: La palabra animal, es un término genérico donde cabe el animal del vecino, el que se tiene en casa, el cual puede ser un perro o un gato, todos son animales, ejemplares de animales, con características comunes y diferentes. Se puede tocar al perro, al gato, pero no al animal en sentido genérico. (Patiño, 2005)

Tanto en bases de datos, como en P.O.O una entidad (clase) es la abstracción o representación mental o gráfica de uno o grupo de objetos, conceptos o ideas del mundo real, entre ellas: estudiante, empleado, artículo, informe, casa, universidad, libro, etc.

La abstracción afrontada desde el punto de vista del Paradigma de la Programación Orientada a objetos identifica las particularidades fundamentales de un objeto, que lo hace diferente a los demás.

En P.O.O una clase es un modelo que representa un conjunto de entidades con características similares y permite abstraer sus datos y operaciones. Dentro de la clase existe otra forma de abstracción cuando se ocultan los atributos, al declararlos de forma privada y luego crearles un método o propiedad de acceso para acceder a ellos desde otras clases. Este proceso es conocido como encapsulamiento.

En el proceso de enseñanza – aprendizaje de los diferentes cursos de P.O.O, el autor ha podido constatar como los estudiantes presentan dificultades en la interpretación de problemas, al planificar su solución, en la elaboración del algoritmo que lo soluciona, en la modelación gráfica de sus componentes esenciales y la identificación de sus relaciones intrínsecas.

Por otra parte, Serna (2011) considera que algunos Ingenieros Informáticos y Licenciados en Ciencias de la Computación son capaces de resolver problemas complejos y diseñar softwares con prontitud, claridad, originalidad y elegancia, mientras que otros no lo pueden lograr, aun cuando hayan recibido los mismos contenidos en sus estudios universitarios

¿Será que no todos los estudiantes han desarrollado el pensamiento abstracto para concebir, planificar y desarrollar tareas complejas, como lo es un software?

¿Se puede, conscientemente desarrollar el pensamiento abstracto en el proceso de enseñanza – aprendizaje de la Programación?

Si bien, por una parte, se hace necesario un pensamiento abstracto avanzado para resolver problemas complejos, el asunto en cuestión es descubrir en qué medida la disciplina Programación, especialmente la P.O.O y las Estructuras de Datos, potencian y favorecen el

desarrollo de la abstracción en educandos y educadores lo que les permitirá afrontar con mayor éxito otros problemas en otras áreas del conocimiento.

En virtud de todo lo expresado es coherente resaltar lo expresado por Cortés (2017), en el que se expresa que la comunidad de Madrid lanzó en 2014 un programa pionero en España que tenía como objetivo convertir la programación en una disciplina curricular en la región, a pesar de que la deficiente formación del claustro de educadores, para impartirla, esté poniendo en duda su eficacia.

Igualmente, Butterfly (2014) presume que aprender a programar contribuye al mejoramiento del desempeño profesional en otras áreas del conocimiento humano debido a que la creatividad, la inventiva o pensamiento original no se limita a la pintura, la música y el teatro. La creatividad se ha extendido progresivamente al arte de la Programación

En el presente artículo, el autor, quien es y ha sido profesor de Programación en la Universidad Tecnológica de la Habana “José Antonio Echeverría” y la Universidad “José Eduardo Dos Santos” en Angola, con el empleo de métodos de investigación científica, entre ellos: el análisis documental, la modelación y el enfoque de sistema, argumenta la necesidad, importancia y factibilidad de formar el pensamiento abstracto en estudiantes y profesores para afrontar con éxito la Programación Orientada a Objetos y dentro de ella el trabajo con las Estructuras de Datos, y recíprocamente, en qué medida los mencionados contenidos desarrollan la abstracción, para emprender problemas intelectuales, laborales y de la vida cotidiana, con éxito.

Desarrollo

Entre los conocimientos y habilidades que debe poseer un Ingeniero Informático, un programador o estudiante de Ciencias de la Computación se encuentra el análisis, creación y trabajo con colecciones de datos, entre ellos: arrays, listas, listas enlazadas, pilas, colas y árboles. Estructuras de datos que implementan los lenguajes de programación más empleados en la actualidad, conocidas también como Tipos de Datos Abstractos (TDA).

La utilidad de emplear arreglos (arrays) está dada en la insuficiencia que tiene una variable simple de solo contener un dato. Si se desea almacenar varios datos sin que uno desplace al

anterior, se necesitarían variables diferentes para lograrlo. Sin embargo, otras limitaciones en los arrays provocaron la construcción de nuevas estructuras de datos: Los arrays dinámicos o listas contiguas y las listas enlazadas.

La solución eficiente de un problema en las disciplinas de Introducción a la Programación, Programación Orientada a Objetos y Estructuras de Datos, con el empleo de los arrays estáticos y dinámicos, las listas enlazadas u otras como los árboles, en la carrera de Ingeniería Informática y otros cursos de Programación en el contexto universitario, ha sido, en ocasiones, una interrogante difícil de responder acertadamente. Cada una de las estructuras antes mencionadas tiene una relación costo - beneficio al crearla, insertar elementos, recorrerla, borrar, buscar un elemento y ordenarla. No existe la estructura perfecta para todas las operaciones.

Supongamos que se desea guardar un conjunto de números enteros para recorrerlos y realizar operaciones sobre ellos. Consideremos que las operaciones serán las siguientes:

1. Almacenar 10 números y mostrarlos (Arrays)
2. Almacenar tantos números como se necesiten, de forma contigua y mostrarlos (Listas contiguas)
3. Almacenar todos los números que se deseen, de forma no contigua y mostrarlos (Listas enlazadas)
4. Almacenar todos los números que queramos y mostrarlo en orden inverso a como los ingresamos. El primero en entrar es el último en salir (Pilas).
5. Almacenar todos los números que se quieran y mostrarlo en el mismo orden que se ingresa. El primero en entrar es el primero en salir (Colas).
6. Almacenar todos los números que se deseen, de forma no contigua para buscar un elemento en el conjunto de la forma más eficiente posible (Árboles).

Por lo expuesto, se identifican seis posibilidades de almacenar números enteros. ¿Será que se obtendrán los mismos resultados si se realizan de un mismo modo? ¿Significa lo mismo almacenar solo 10 números que realizar una lista contigua indefinida o array dinámico? ¿Cómo se crea una lista no contigua o lista enlazada?

La solución de este problema exige de diferentes niveles o tipos de abstracciones pues se debe separar, dividir en partes cada solución conforme a lo que se solicita, aun cuando la tarea es sustancialmente la misma, almacenar números enteros, se efectuará en estructuras distintas, en diferentes Tipos de Datos Abstractos (TDA). Lo cual significa, inicialmente, el dominio de 6 conceptos para la creación e impresión de datos. A continuación, se facilita una definición operativa para las estructuras de datos más empleadas:

Arrays: Estructura donde se guarda una colección finita de datos del mismo tipo. Se accede a cada elemento individual del array mediante un número entero denominado índice.

Listas contiguas: En caso de que la lista sea no enlazada, entonces se comporta similar a un array, y por lo tanto es una estructura o lista contigua donde los elementos se almacenan en posiciones consecutivas de memoria. Se corresponden con el concepto de array de la mayoría de lenguajes de programación.

Listas enlazadas: Secuencias de nodos que se enlazan con apuntadores o referencias a direcciones de memoria. Una sucesión de nodos en la que a partir de uno se puede acceder al que ocupa la siguiente posición en la lista.



Figura 1. Representación de una lista doblemente enlazada.

Fuente: Elaboración propia.

Pilas: Listas o estructuras de datos en la que el modo de acceso a sus elementos es de tipo LIFO (del inglés Last In First Out, último en entrar, primero en salir) que permite almacenar y recuperar datos.

Colas: Estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción push se realiza por un extremo y la operación de extracción pop por el otro. También se le llama estructura FIFO (del inglés First In First Out), debido a que el primer elemento en entrar será también el primero en salir.

Árboles binarios: Estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos (de ahí el nombre "binario").

¿Cómo una persona se representa mentalmente el almacenaje de los números en cada una de las estructuras previamente identificadas? ¿Dónde se almacenan? ¿Por cuánto tiempo? ¿Cómo se puede acceder a ellas?

Según Rosental y Ludin (1973) *“Las representaciones mentales son formas simbólicas de reproducir la realidad en ausencia de ésta. Para dar sentido a aquello que nos rodea las representaciones se organizan en estructuras y se construyen partiendo de un contexto social propio de un grupo determinado. Es decir, el ser humano forma representaciones mediante una serie de intercambios o interacciones con los demás”* (p.2).

Los citados autores identifican como ejemplos de representación mental los esquemas, símbolos, imágenes, ideas y otras formas de pensamientos.

El ser humano estructura representaciones mediante intercambios o interacciones con los demás, con objetos, procesos y fenómenos que ocurren en su diario quehacer. El pensamiento abstracto tiene un notable sustento en las representaciones mentales del diario vivir.

Cuando se trabaja con arrays, listas y árboles, tipos de datos abstractos que no se palpan, ni se perciben en el interior de la computadora al ser creados, recorridos y borrados, en este proceso cada estudiante, acto seguido de abstraerse mentalmente, teniendo en cuenta las características y conceptualización de las identificadas estructuras, concibe su particular representación mental, que le permite modelarlo gráficamente para su posterior discernimiento.

La previa representación mental y posterior modelación gráfica son partes del proceso de abstracción que se desarrolla al trabajar los tipos de datos abstractos en la asignatura Programación.

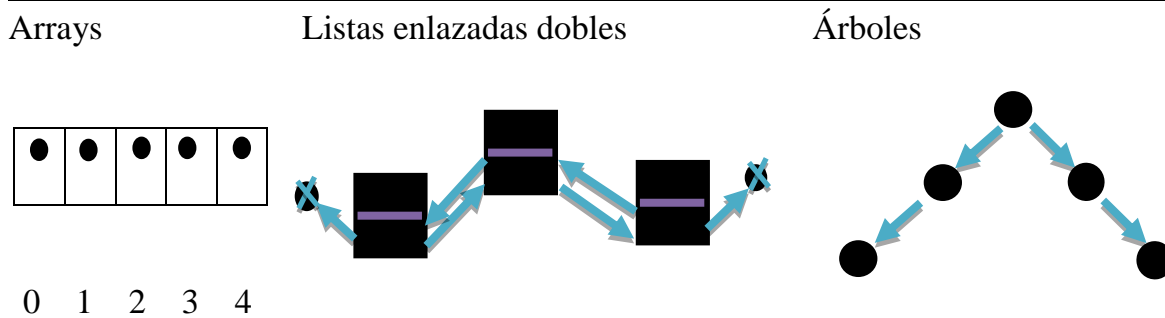


Figura 2. Representación de: Arrays; Listas enlazadas y Árboles.

Fuente: Elaboración propia.

En cualquiera de las tres estructuras representadas se pueden almacenar el conjunto de números enteros del ejercicio anterior, lo importante es identificar el más eficiente para cada caso.

A continuación, el autor expone un conjunto de ideas que constituyen recomendaciones para el desarrollo del pensamiento abstracto.

Recomendaciones didácticas y metodológicas que propiciarán el desarrollo del pensamiento abstracto en los estudiantes a través del aprendizaje de la P.O.O y las Estructuras de Datos.

1. Lee detenidamente el problema, examen o ejercicio planteado e identifica las clases o entidades que intervienen en el asunto, sus atributos y acciones a realizar sobre cada una de ellas, compáralas y revela sus posibles nexos.
2. Descubre las similitudes que existen entre el problema, examen o ejercicio planteado en clase con el quehacer cotidiano. Haz memoria de algún escenario análogo.
3. Realiza la representación mental de las entidades comprometidas en la solución del problema y sus posibles relaciones, tantas veces como sea necesario hasta su total discernimiento. Imagínalas y visualízalas en tu mente.
4. Dibuja, representa gráficamente cada entidad (clase) con todos sus atributos, acciones a realizar y sus relaciones o vínculos.
5. Algoritmiza, modela gráfica o teóricamente, o de ambas formas, los pasos para solucionar cada tarea requerida en el problema.
6. Identifica la estructura de datos más eficiente para solucionar el problema, entre ellas: arrays, listas contiguas, listas enlazadas, pilas, colas, árboles y grafos. Compáralas. Imagínala

y visualízala en tu mente. Haz conjeturas de cómo se almacenan los datos en la computadora en cada estructura.

7. Dibuja, representa gráficamente la estructura de datos seleccionada y asóciala con el quehacer diario. Imagínala creada en la memoria de la computadora.

8. Elabora el código de programación, paso a paso y ejecútalo con diversos juegos de datos para comprobar la solución del problema.

9. Razona tus ideas y soluciones con otros estudiantes, descubre otros puntos de vistas y confróntalos con los tuyos.

Otras recomendaciones que propician el desarrollo del pensamiento abstracto.

1. Elabora mapas conceptuales: representaciones gráficas para organizar e interpretar el conocimiento. Contienen conceptos, y relaciones entre conceptos.

2. Define conceptos de objetos o procesos. Expresa de manera exacta y precisa el significado de algo y represéntalo gráficamente.

3. Dado un esquema o un diseño de un objeto o proceso, descríbelo, expresa su significado y reconstrúyelo diferente al original.

4. Proyecta por adelantado la solución de alguna situación problemática en el orden docente, social, laboral o personal, antes que otras personas o instituciones y confronta la solución con la realidad.

5. Elabora estrategias y modelos para solucionar problemas.

6. Realiza el resumen de un texto, por ejemplo, resumir en 10 renglones lo expresado en dos cuartillas de un libro.

Conclusiones

El pensamiento abstracto es una facultad del ser humano que posibilita la creación de ideas, imaginar sucesos, planificar soluciones y proyectarse al futuro.

El desarrollo del pensamiento abstracto se encuentra íntimamente relacionado con la con la representación mental y modelación de objetos, procesos y fenómenos de la realidad o creados en la imaginación.

Para enfrentar con éxito el proceso de enseñanza – aprendizaje de la Programación Orientada a Objetos y las Estructuras de Datos, se precisa de haber adquirido un determinado nivel abstracción y recíprocamente este se alcanza y desarrolla transitando por dicho proceso.

El empleo de objetos reales o modelos de objetos, fotografías, dibujos, diagramas, símbolos, grabaciones de audios y otros sonidos y los mapas conceptuales, en las clases de Programación, contribuyen al desarrollo del pensamiento abstracto en educadores y educandos.

El aprendizaje de la Programación, tarea compleja y retadora, en donde se escriben instrucciones que un ordenador ha de ejecutar para resolver un problema, propicia, entre otras cosas, el desarrollo del pensamiento lógico y abstracto.

Numerosas investigaciones en diversos países han considerado a la programación como la gran disciplina del presente y el mañana que prepara al ser humano para trabajar en equipos, afrontar desafíos como la resolución de problemas complejos, la gestión de conflictos y la elaboración de proyectos.

Referencias bibliográficas

NaranJo, L.M. y Puga L. A. (2016). El pensamiento lógico – abstracto como sustento para potenciar los procesos cognitivos en la educación. *Colección de Filosofía de la Educación*, No. 21. Recuperado de: www.redalyc.org/articulo.oa?id=441849209001.

Flores, A. (2011). Desarrollo del Pensamiento Computacional en la Formación en Matemática Discreta. *Lankpsakos*, No. 5. Recuperado de: <http://www.funlam.edu.co/revistas/index.php/lampsakos/article/view/815/783>.

Krivoy, A. (2004) Maimónides. El pensamiento abstracto y las causas de sus dificultades. *Sociedad médica*, Volumen 49, No.1. Recuperado de: <https://www.revistacentromedico.org/ediciones/2004/1/art-9/>

Serna, M. E. y Polo, J. A. (2014). Lógica y abstracción en la formación de ingenieros: una relación necesaria. *Ingeniería, Investigación y Tecnología, Volumen 15*, No 2. Recuperado de <https://www.sciencedirect.com/science/article/pii/S1405774314722188>.

Rosental, M. y Ludin P. (1973). *Abstracción*. Diccionario filosófico. La Habana. Cuba: Editorial Política.

Patiño, J.I. (2005). Programación Orientada A Objetos en Lenguajes no orientados a objetos: C, una experiencia. *Scientia Et Technica, Vol. XI, No. 29*. Recuperado de <https://www.redalyc.org/pdf/849/84911948018.pdf>

Serna, E. (2011). La abstracción como componente crítico de la formación en ciencias computacionales. *Avances en Sistemas e Informática, Vol.8, No.3*. Recuperado de <https://www.sciencedirect.com/science/article/pii/S1405774314722188>

Castellaro, M. (2011). El concepto de representación mental como fundamento epistemológico de la Psicología. *Límite. Vol.6, No.24*. Recuperado de <https://www.redalyc.org/pdf/836/83622474005.pdf>

Rosental, M. y Ludin, P. (1973). *Representación*. Diccionario filosófico. La Habana. Cuba: Editorial Política.

Soler, Y. y Lezcano, M. G. (2008). Organización del conocimiento de la asignatura Estructuras de datos y algoritmos para ingeniería informática basada en mapas conceptuales. *Avances en Sistemas e Informática, Vol.5, No.3*. Recuperado de <https://www.redalyc.org/articulo.oa?id=133117550017>.

Martínez, L. (2017). Desarrollo cognitivo y educación formal: análisis a partir de la propuesta de L. S. Vygotsky. *Universitas Philosophica, Vol.34, No.69*. Recuperado de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-53232017000200053

Kusmaryono, I. y Dwijanto D. (2018). Analysis of abstract reasoning from grade 8 students in mathematical problem solving with solo taxonomy guide. *Infinity, Volumen 7, No. 2*. Recuperado de: <https://www.researchgate.net/publication/>.

Joãos M., Gonçalves, L. y Sampaio, J. (2012). Do concreto para oabstrato pelo uso dos sentidos e de sensores: um estudo de caso sobre o ecossistema poças de maré. *Investigação & Inovação em Educação, Vol.2, No.2*. Recuperado de <https://core.ac.uk/download/pdf/47140996.pdf>.

Barría, C. (2018, 7 de febrero). Cómo es el "Método Singapur" con el que Jeff Bezos les ha enseñado matemáticas a sus hijos (y por qué lo usan los mejores estudiantes del mundo). *BBC Mundo*. Recuperado de: <https://www.bbc.com/mundo/noticias-42966905>. p. 2-4.

Butterfly, A. (2014, 7 de septiembre). Por qué vale la pena aprender a programar... ¡nunca es tarde! *BBC Mundo*. Recuperado de: <https://www.bbc.com/mundo/noticias/2014/09/.p.1-3>

Cortés,J. (2017, 6 de abril). Educación. La importancia de que los niños aprendan a programar. *El País*. Recuperado de: <https://retina.elpais.com/retina/2017/04/03/innovacion/.html>. p.1